**UNIVERSIDAD DE ORIENTE.**

**NUCLEÓ ANZOÁTEGUI.**

**ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS**

**DEPARTAMENTO DE INGENIERIA.**

****

|  |  |  |
| --- | --- | --- |
| TUTOR | PROFESOR: | ING. ALFONSO ALFONSI |

EVALUACIÓN 1: ASIGNACIÓN ESQUEMA DE ENTRADA Y SALIDA, PWM, ENTRADA DE POTENCIOMETRO, INTERRUPCIONES, TEMPORIZADORES

|  |  |
| --- | --- |
| CORREA LUIS | CARDOZA MARIA |
| CI N° 19840230 | CI N° 6142718 |
| ASIGNATURA: ARQUITECTURA AVANZADA DEL COMPUTADOR | SECCIÓN: 01 |

**BARCELONA, 14 DE ABRIL DE 2015**

CONTENIDO

1. INTRODUCCIÓN 3

2. DESARROLLO (EXPERIENCIAS, DIAGRAMAS DE LOS CIRCUITOS Y CÓDIGOS) 3

3. DISCUSIÓN DE RESULTADOS 8

4. CONCLUSIONES 8

5. REFERENCIAS 8

6. NORMAS 8

## INTRODUCCIÓN

En el presente trabajo se realizan la carga de los programas en el hardware denominado Arduino Uno[[1]](#footnote-1) y se verifican los circuitos:

1. El esquema de entrada y salida digital
2. Salida de PWM.
3. Entrada de potenciómetro
4. Interrupciones
5. Temporizador

El presente trabajo muestra los esquemas y algoritmos utilizados para realizar la asignación.

## DESARROLLO (EXPERIENCIAS, DIAGRAMAS DE LOS CIRCUITOS Y CÓDIGOS)

a) Entrada y salida

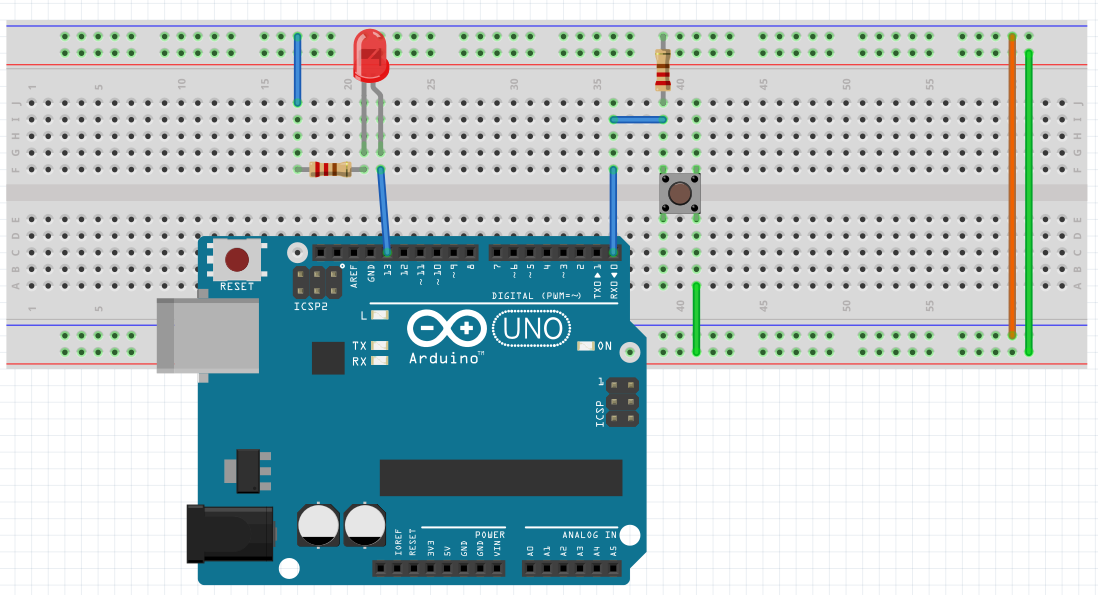


Figura N° 1. Entrada y Salida de Arduino Uno.

int entrada = 1;

int salida = 10;

void setup(){

pinMode(entrada, INPUT);

pinMode(salida,OUTPUT);

}

void loop(){

int valor = digitalRead(entrada);

if(valor!=0){

digitalWrite(salida,HIGH);

}else{

digitalWrite(salida,LOW);

}

}

b) Salida PWM

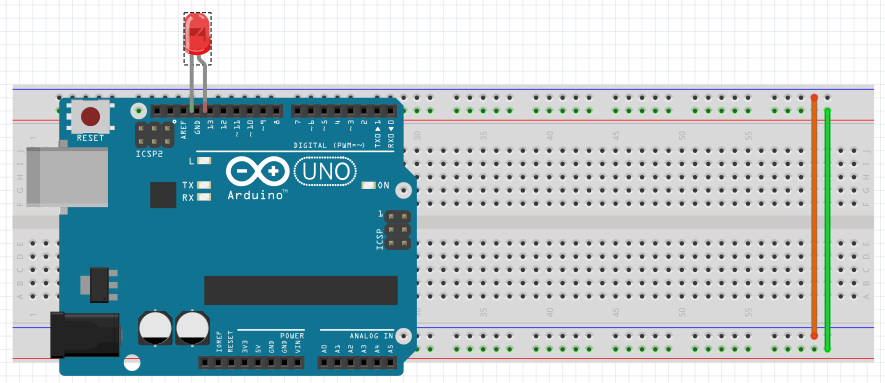


Figura N° 2. Salida PWM de Arduino Uno.

int digPin = 13; // pin digital 13

void setup() {

pinMode(digPin, OUTPUT); // pin en modo salida

}

void loop(){

digitalWrite(digPin, HIGH); // asigna el valor HIGH al pin

delay(1000); // espera medio segundo

digitalWrite(digPin, LOW); // asigna el valor LOW al pin

delay(1000); // espera medio segundo

}

c) Entrada de potenciómetro:

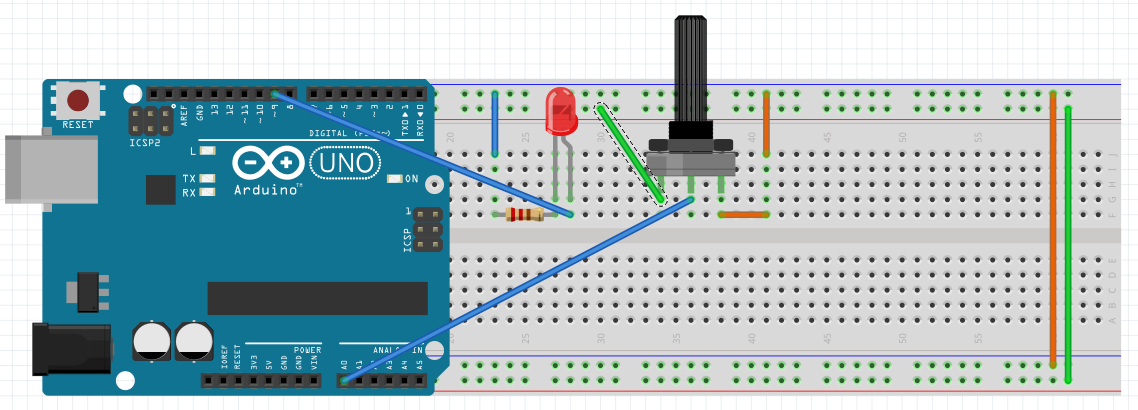


Figura N° 3 Entrada Potenciómetro Salida PWM.



Figura N° 4 Modulación PWM.

const int pot=0;//se declaran como entrada automaticamente

const int led = 3;

void setup(){

pinMode(led,OUTPUT);

}

void loop(){

int brillo=analogRead(pot);

int brilloled = map(brillo, 0, 1023, 0, 255);

analogWrite(led,brilloled);

}

d) Interrupciones:

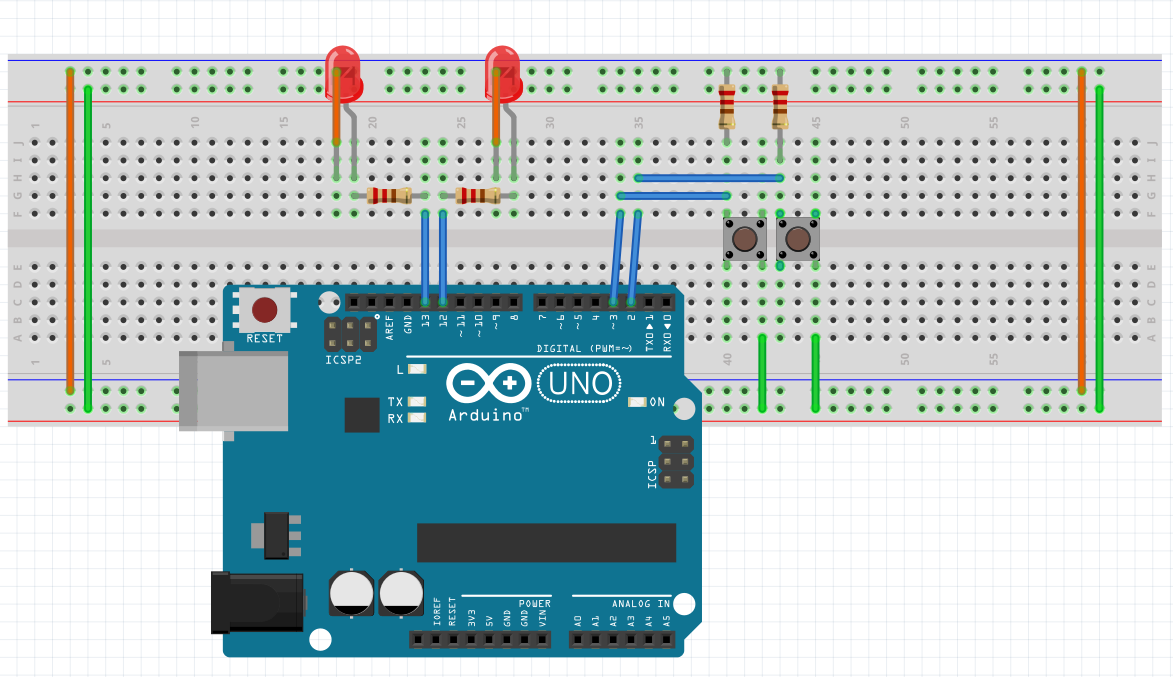


Figura N° 4. Interrupciones Arduino Uno.

int pin = 13;

volatile int estado = LOW;

//el atmega328p solo tiene 2

// el 0 pin digital 2, el 1 en el pin digital 3

void setup(){

pinMode(pin, OUTPUT);

attachInterrupt(0, parpadeo, CHANGE);//CHANGE CAMBIO

}

void loop(){

digitalWrite(pin, estado);

}

void parpadeo(){

estado = !estado;

}

e) Temporizador.

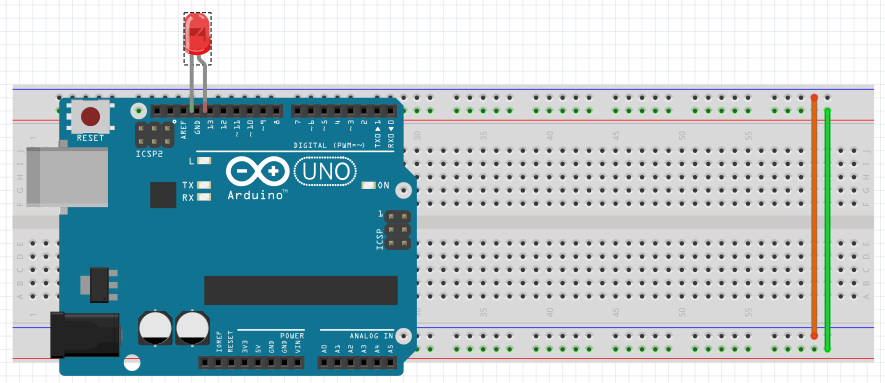


Figura N° 5. Temporizadores en Arduino Uno.

int led1 = 13;

void setup() {

pinMode(led1, OUTPUT);

digitalWrite(led1, LOW);

}

void loop() {

digitalWrite(led1, HIGH);

delay(1000);

digitalWrite(led1, LOW);

delay(1000);

}

## DISCUSIÓN DE RESULTADOS

* 1. En los resultados se pudo comprobar el uso del Arduino uno al utilizarlo con switches, potenciómetros, sus salidas PWM y sus dos interrupciones.

COMPONENTES UTILIZADOS:

Arduino Uno

Potenciómetro

Resistencias

Leds

Switches

Cables

Software: Arduino Ide, Fritzing.

ACCIÓN Y RESULTADO:

Se enviaron señales a través del Arduino y el mismo envió señales de respuesta

* 1. Utilice el método delay en el Arduino ide para el control del sensor ultrasónico.

COMPONENTES UTILIZADOS:

Arduino Uno

Resistencias

Leds

Cables

Sensor Ultrasonico

Software: Arduino Ide, Fritzing.

ACCIÓN Y RESULTADO:

Se utilizó el sensor ultrasónico HC-SR04 para determinar la existencia de un objeto cerca del Arduino Uno.

## CONCLUSIONES

* Se comprueba el funcionamiento del Arduino uno al enviar señales digitales en respuesta a otras entradas digitales.
* Se comprueba el funcionamiento del Arduino uno al enviar señales pwm para prender y apagar un led.
* Se comprueba el funcionamiento del Arduino uno al enviar señales pwm en respuesta a una señal recibida de un potenciómetro
* Se comprueba el funcionamiento del Arduino uno que cuenta con dos interrupciones externas.
* Se comprueba el funcionamiento del Arduino uno al poder controlar el tiempo de envió de señales con el método delay.

## REFERENCIAS

* Óscar Torrente Artero., (2013). Arduino Curso Práctico de formación.

## NORMAS

* Normas Apa: Para redacción de documentos

1. Arduino se inició en el año 2005 como un proyecto para estudiantes en el [Instituto IVREA](http://en.wikipedia.org/wiki/en:Interaction_Design_Institute_Ivrea), en [Ivrea (Italia)](http://es.wikipedia.org/wiki/Ivrea_%28Italia%29) [↑](#footnote-ref-1)